Improvements in Beam Search

Volker Steinbiss, Bach-Hiep Tran, Hermann Ney[1]

Philips GmbH Forschungslaboratorien

Weisshausstr. 2 • 52066 Aachen • Germany • E-mail: tran@pfa.philips.de

Abstract

Time-synchronous beam search has successfully been employed in the Philips continuous-speech recognizer for several years now, handling a vocabulary of 20 000 words and more. We have now improved the search procedure with two robust pruning methods that drastically reduce both average and peak search effort.

1. Introduction

In the Philips' research prototype for large-vocabulary continuous-speech recognition, the time-synchronous beam search for the most likely word sequence accounts for the lion's share of memory demand and for a large part of computation. This paper describes two methods which have proven to reduce both peak and average size of the active search space and to be more robust than our standard pruning method.

All investigations are within the context of our recognizer which is described elsewhere in enough detail ([7, 9]) and which can briefly be characterized as follows: It is based on the statistical approach, is phoneme-based and uses Hidden Markov models (HMMs) consistently with the Viterbi approximation, with non-tied continuous mixture densities. We use trigram and (typically) bigram language models. While the system has been used under various conditions up to 45 000 words (with the American English Wall Street Journal task [9]), the experiments reported here refer to the recognition of German real-life dictations [9] on analog desk-top equipment with a vocabulary of 15 000 words.

2. The Baseline Search Procedure

Before we explain the new methods, let us shortly review and comment on the search procedures currently used in our system [3].

Nomenclature. A state in the search space at a certain time is called point or grid point. As a point during search contains information about a specific partial sentence hypothesis, the terms *point*, *grid point* and *hypothesis* are used interchangeably.

Time-Synchronous Breadth-First Search. Search processes one observation (centisecond frame) after the other. All active hypotheses refer to the same input which facilitates the comparison of hypotheses. [6]

Data-Driven Beam Search. At each time frame, only hypotheses with a score relatively close to the best hypothesis are considered further. The beam width is defined by a pre-defined "pruning threshold". [6]

Tree Organization. In contrast to linear search, where each HMM state belongs to exactly one word, we use a tree organization that takes advantage of the fact that many words share the same initial phoneme sequence [5].

Forest Search. In time-synchronous beam search, all knowledge sources are expanded ("multiplied out") into one network. In particular, there are separate tree copies due to the language model constraints. [5]

Phoneme Look-Ahead. A look-ahead six frames ahead of the current frame further reduces the search space [2, 5].

Language Model Pruning. An additional reduction is achieved with an additional pruning step only regarding word ending states.

Other. The construction of a word graph or lattice for rescoring with a higher-order LM [1, 4] or for speech understanding [8] is not considered here.

Search Errors (Pruning Errors). If, due to non-exhaustive search, the search procedure fails to find the optimal state sequence - optimal with respect to the knowledge sources, not necessarily associated with the spoken word sequence - this is called a search error. As an alternative to directly determining search errors, we directly observed the word-error rate: If the errors change in connection with a search space reduction, a search error has occurred.

[1] Lehrstuhl für Informatik VI, University of Aachen (RWTH), 52056 Aachen, Germany. E-mail: ney@informatik.rwth-aachen.de

3. HISTOGRAM PRUNING

In our off-line recognition experiments, we frequently observed that the peak search effort exceeded the average effort by almost two orders of magnitude (Table 1). This behaviour is typically observed during non-speech sounds and hesitations.

Table 1: Maximal versus average number of grid points per centisecond before pruning.

Speaker	M-60	M-61	M-64
Maximal	1 287 000	717 000	3 376 000
Average	20 000	20 000	44 000
Ratio max./av.	64	36	77

Table 2: Maximal number of grid points per centisecond before pruning for three speakers.

Upper Limit	M-60	M-61	M-64
10 000	18 000	15 800	19 200
25 000	51 300	50 600	45 000
50 000	96 000	91 000	83 100
100 000	176 800	162 900	166 600
150 000	246 200	231 300	227 900
200 000	320 100	289 600	292 500
none	1 287 600	717 000	3 375 500

A straightforward approach to the problem[1] is to introduce an additional pre-specified upper limit on the number of active points (per frame). With a histogram on the hypotheses scores[2] of a specific time frame, the pruning threshold is decreased, if necessary, to keep the number of active hypotheses below this limit. Between expansion of hypotheses of the preceding time frame and pruning of the set of expanded hypotheses, the number of active points may well exceed the given limit. Tables 2 and 3 show the dependence of search effort on the given limit. E. g. for speaker M-60 and a limit of 100 000 points per centisecond, there are 16 000 points active on average, 176 800 is the maximal value before and, by construction, 100 000 the maximal value after pruning. M-64 was chosen as a particularly bad speaker.

Table 3: Average number of grid points per centisecond before pruning.

Upper Limit	M-60	M-61	M-64
10 000	1 200	1 100	8 300
25 000	12 000	13 000	16 000
50 000	14 000	17 000	22 000
100 000	16 000	20 000	28 000
150 000	17 000	20 000	30 000
200 000	19 000	21 000	32 000
none	20 000	20 000	44 000

The interesting question is how many additional search errors are introduced by this new pruning scheme. Quite astonishingly, we could reduce the upper limit on the points quite drastically without detecting search errors. Table 4 and an inspection of the global sentence scores indicate that search errors occur and deteriorate performance around and below 25 000 points maximum. **For the subsequent experiments, we thus fixed 30 000 points as upper limit.**

Table 4: Word-error rate in %.

Upper Limit	M-60	M-61	M-64
10 000	24.8	36.7	25.2
25 000	11.5	13.8	22.8
50 000	11.6	13.7	22.1
100 000	11.6	13.7	21.8
150 000	11.6	13.7	21.8
200 000	11.6	13.7	21.6
none	11.6	13.7	21.4

Another nice feature of histogram pruning is robustness. If pruning is too tight, pruning errors are likely to occur. If we compare the relation between the number of processed points and errors due to pruning both for standard beam pruning and (beam pruning with) histogram pruning, the latter compares very favourably: E. g. for a certain speaker (M-72), reducing the average search effort of standard pruning from 20 000 to 3 000 points more than doubles the error rate from 11.1% to 36.9%, while histogram pruning with 3 000 points remains with 14.1% (a quarter more) relatively stable.

We conclude that the peaks of about 1 million points per frame are drastically reduced by histogram pruning to only 30 000, the average number by more than 30%, without deterioration of the error rate. In addition, the degradation due to incorrectly chosen parameters is more graceful than with the conventional beam pruning.

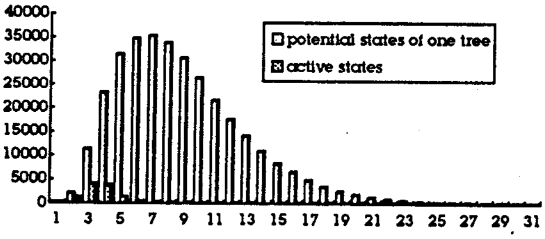

Fig. 1: Distribution of states over generations (corresponding to phoneme positions) of a tree: Comparison of total search effort using (histogram) pruning with potential search space in only one *tree.*

[1] Gerhard Bachmayer (Philips Dictation Systems, Vienna) indicated the problem to us.

[2] In this paper, negative log probabilities that occur during recognition are called "scores".

Fig. 1 gives an impression of how the active points are distributed. Compared to the potential search space, which is a multiple (here 15 000) of the number of states in *one* tree, the active search is more concentrated on the first generations (i. e. the word beginnings). Typically 10-15 trees are active on the average.

4. LANGUAGE-MODEL LOOK-AHEAD: SMEARING LM SCORES OVER THE TREE

4.1 Motivation

The efficient method of forest search (or tree search; cf. section 2) basically consists of structuring the vocabulary in a phonetic tree and introducing tree copies due to the language model constraints. The word identities are only known at the word endings rather than at the word beginnings in linear search, such that the LM (language model) knowledge is employed with one word delay. Instead, it should be incorporated as early as possible.

Another deficiency lies in the fact that the scores of hypotheses change drastically when a word ending is encountered because the LM scores are added there. In particular, the pruning threshold has to exceed largest LM score. If some of the LM scores are close to the pruning threshold, pruning errors are likely to occur. If, on the other hand, a word ends with a good score, many useless word start hypotheses are being examined during silence after a word.

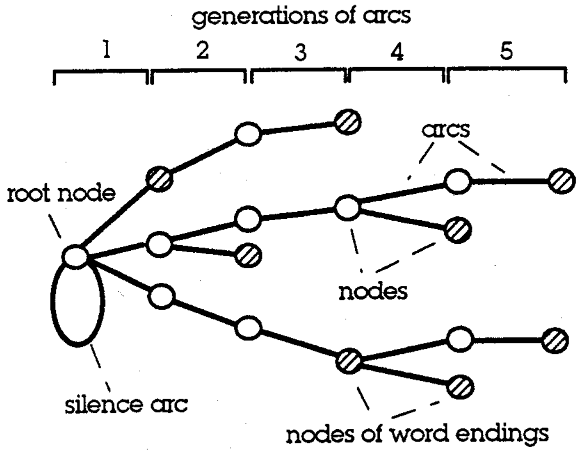

Fig. 2: Sample tree for illustration of nomenclature.

4.2 Description of LM Look-Ahead

The basic idea of our method is to incorporate the LM scores as early as possible: In each portion of a phonetic tree, an estimate of the language model (LM) scores of all possible word continuations is used for a modified pruning strategy. We describe the basic algorithm first and later-on simplified variants that can be more efficiently implemented.

Nomenclature: Every tree consists of nodes and arcs. Every arc corresponds to a phoneme copy or a sequence of typically six HMM states. Leaves of the tree always are word ending nodes, but word ending nodes also occur within the tree.

For every tree copy, representing the LM left context, and every word ending node, there is a LM score. The algorithm modifies the usual time synchronous beam search as follows:

```
Pre-processing before recognition:
For every tree and every node n do:
  calculate and store MinLMScToExpect(n)
  := min {LM score of all nodes n' that
  can be reached from n};
end of loop;
Search during recognition:
For every state being expanded:
  if a state is expanded from a predeces-
  sor arc ending with node n' to an arc
  ending with node n (n'≠n), add MinLMSc-
  ToExpect(n)-MinLMScToExpect(n') to the
  score;
  if a word ending is reached in node n,
  subtract MinLMScToExpect(n) while add-
  ing the usual LM-score.
end of loop;
```

It should be noted that in a word-ending node, *MinLMScToExpect* can be smaller than the LM score.

As the method described above requires either a large amount of storage (or computation, if the values are calculated on-the-fly during recognition), several simplifications should be considered (we assume an n-gram model with n>1). E. g., **in the subsequent experiments, we only use the unigram LM probablilities for the LM look-ahead ("unigram approximation")**, thus reducing the storage (or computation) effort. Various other simplifications are possible.

Rather than emphasizing the look-ahead, the method can also be viewed as smearing the LM scores over the tree rather than concentrating them in the word endings.

4.3 Experimental Results

We used the language-model look-ahead derived from a unigram model; control experiments with more sophisticated set-ups showed only slightly better results. In addition to the histogram pruning method, the average search space size is reduced *by* more than 70%, e.g. from 20 000 to 5 000 active hypotheses per frame. Table 5 and Fig. 3 show the search effort broken down into arc generations: An arc in generation *g* corresponds to a phoneme in position *g*. They indicate that the search space is reduced and more concentrated in the word beginnings. Here, the same small amount of search errors was allowed both without and with LM look-ahead. The improved pruning is partly due to the modified relation between within-word phonemes and silence.

Table 5: Effect of LM look-ahead on the number of active grid points in the generations.

Generation	No LM-LA	LM-LA	Gain
1	166	254	0.65
2	1 241	657	1.89
3	4 039	665	6.07
4	3 709	426	8.70
5	1 360	214	6.35
6	390	97	4.02
7	137	44	3.1
8	53	20	2.7
9	22	10	2.2
10	10	5	2
...	...	...	...
Total:	11 141	2 406	4.63

The robustness of this method is quite impressive. Again for speaker M-72, while reducing the average search space of standard pruning from 20 000 to 3 000 points more than doubles the error rate of 11.1%, there is no significant change with LM look-ahead for 3 000 points and even for a reduction to 900 points only a slight increase to 12.2%!

The last Fig. 4 shows the search space size for the standard and the new methods over time.

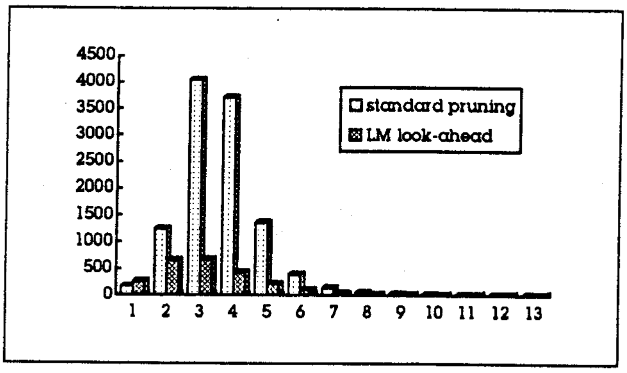

Fig. 3: Distribution of search effort in different generations (grid points in the 1st up to 13th phoneme) for baseline system and LM look-ahead. Cf. Table 5.

5. SUMMARY

Histogram pruning and language-model look-ahead pruning improve large-vocabulary search, significantly reducing peak and average search space size and thus both memory and computation. At the same time, they are more robust than standard beam pruning.

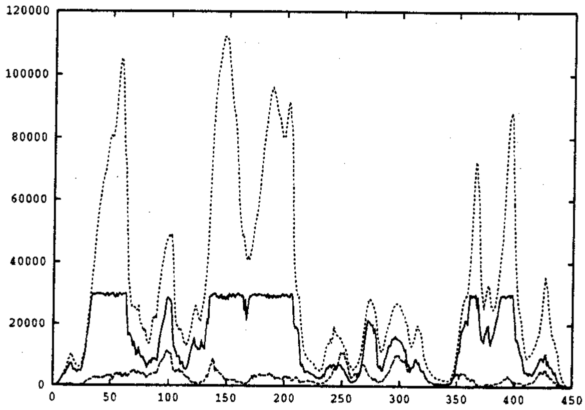

Fig. 4: Search space size over time (in centisecond frames) for baseline method (dashed), histogram pruning (solid line) and LM look-ahead (dotted).

ACKNOWLEDGEMENT

The work was partly sponsored by the German Federal Ministry for Research and Technology (BMFT) under grant No. 01 IV 102 B3. Only the authors are responsible for the contents of this publication.

REFERENCES

Abbreviation: *ICASSP* stands for *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing.*

[1] X. Aubert, C. Dugast, H. Ney and V. Steinbiss: "Large Vocabulary Continuous Speech Recognition of Wall Street Journal Data", ICASSP, Adelaide, pp. II-129 - 132, April 1994.

[2] R. Haeb-Umbach and H. Ney: "A Look-Ahead Search Technique for Large-Vocabulary Continuous-Speech Recognition", Proc. Europ. Conf. on Speech Communication and Technology, Genova, pp. 495-498, Sep. 1991.

[3] Ney H.: "Search Strategies for Large-Vocabulary Continuous-Speech Recognition", Proc. of NATO Advanced Study Institute on Speech Recognition and Understanding, Bubion, Spain, 1993, in press.

[4] H. Ney, X. Aubert: "A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition", elsewhere in these Proceedings.

[5] H. Ney, R. Haeb-Umbach, B.-H. Tran and M. Oerder: "Improvements in Beam Search for 10000-Word Continuous Speech Recognition", ICASSP, San Francisco, CA, pp. I-9 - I-12, March 1992.

[6] H. Ney, D. Mergel, A. Noll and A. Paeseler: "Data Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition", IEEE Trans. on Signal Processing, Vol. SP-40, No. 2, pp. 272-281, Feb. 1992.

[7] H. Ney, V. Steinbiss, R. Haeb-Umbach, B.-H. Tran and U. Essen: "An Overview of the Philips Research System for Large-Vocabulary Continuous-Speech Recognition", *to appear in* Int. Journal of Pattern Recognition and Artificial Intelligence, 1994.

[8] Oerder M. and Ney H.: "Word Graphs: An Efficient Interface Between Continuous-Speech Recognition and Language Understanding", Proc. ICASSP'93, Minneapolis, MN, pp 119-122, 1993.

[9] V. Steinbiss, H. Ney et al.: "Continuous Speech Dictation - From Theory to Practice", in preparation.